Amendments to the Claims:

Claims 1, 11, 20, 30, & 39 have been amended as per Examiner's request to further clarify the metes and bounds of "N devices". Further explanation is presented in Remarks section; no new matter has been entered.

Claims 3, 13, 22, 32, & 41 are dependent upon amended base claims 1, 11, 20, 30, & 39, respectively, and make reference to the further limitation that the existing task file is specifically an IDE task file. No new matter is entered.

Claims 4, 14, 23, 33, & 42, being previously dependent upon claims 3, 13, 22, 32, & 41, have been amended to being dependent upon claims 1, 11, 20, 30, & 39. No new matter is entered.

Claim 10 has been amended to correct a minor typographical error: "corresponding the" has been amended as "corresponding to the". No new matter is added.

Listing of Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended) An electronic system comprising:

a host;

a controller electrically coupled to the host through a single port of a predetermined interconnection means, the ~~single port~~ predetermined interconnection means being designed for providing the host access to a maximum of N devices; and

M peripheral devices electrically coupled to the controller;

wherein M is greater than N and the controller allows the host to access the peripheral devices using the single port, and

the host modifies predetermined existing fields in packets or registers in an existing task file that are sent to the controller through the single port to specify a target peripheral device.

2. (original) The electronic system of claim 1, wherein the predetermined interconnection means is an Integrated Drive Electronics (IDE) bus or a Serial AT Attachment (SATA) interface.

3. (currently amended) The electronic system of claim 1, wherein the ~~host modifies predetermined fields in packets or registers in~~ the existing task file is an IDE task file ~~that are sent to the controller through the single port to specify a target peripheral device~~.

4. (currently amended) The electronic system of claim ~~3~~ 1, wherein the predetermined fields are control codes or reserved vendor-specific bits in ATA Packet Interface (ATAPI) packets that are sent to the controller through the single port to specify the target peripheral device.

5. (original) The electronic system of claim 1, wherein the M peripheral devices electrically coupled to the controller at least comprise an optical storage device and a non-volatile storage device.

6. (previously presented) The electronic system of claim 5, wherein the non-volatile storage

device is a flash card access device or a hard-disk drive.

7. (original) The electronic system of claim 1, wherein the host schedules packets sent to the M peripheral devices according to a priority ranking.

8. (original) The electronic system of claim 7, wherein the priority ranking is a dynamic ranking that varies according to operations or speed settings of the peripheral devices.

9. (original) The electronic system of claim 1, wherein the M peripheral devices include a first peripheral device and a second peripheral device, and the controller directly transfers data stored on the first peripheral device to the second peripheral device without buffering the data in the host.

10. (currently amended) The electronic system of claim 1, wherein the host determines which peripheral devices are coupled to the controller and builds a set of virtual drives in an operating system (OS) of the host corresponding to the peripheral devices coupled to the controller.

11. (currently amended) An electronic system comprising:

a host;

a controller electrically coupled to the host through a single port of a predetermined interconnection means, the ~~single port~~ predetermined interconnection means being designed for providing the host access to a maximum of N devices; and

M peripheral devices electrically coupled to the controller, the peripheral devices including a first peripheral device and a second peripheral device;

wherein M is greater than N,

the controller allows the host to access the peripheral devices using the single port,

the host modifies predetermined existing fields in packets or registers in an existing task file that are sent to the controller through the single port to specify a target peripheral device, and

the controller directly transfers data stored on the first peripheral to the second peripheral device without buffering the data in the host.

12. (original) The electronic system of claim 11, wherein the predetermined interconnection means is an Integrated Drive Electronics (IDE) bus or a Serial AT Attachment (SATA) interface.

13. (currently amended) The electronic system of claim 11, wherein the ~~host modifies predetermined fields in packets or registers in~~ existing task file is an IDE task file ~~that are sent to the controller through the single port to specify a target peripheral device~~.

14. (currently amended) The electronic system of claim ~~13~~ 11, wherein the predetermined fields are control codes or reserved vendor-specific bits in ATA Packet Interface (ATAPI) packets that are sent to the controller through the single port to specify the target peripheral device.

15. (original) The electronic system of claim 11, wherein the M peripheral devices electrically coupled to the controller at least comprise an optical storage device and a non-volatile storage device.

16. (original) The electronic system of claim 15, wherein the non-volatile storage device is a

flash card access device or a hard-disk drive.

17. (original) The electronic system of claim 11, wherein the host schedules packets sent to the M peripheral devices according to a priority ranking.

18. (original) The electronic system of claim 17, wherein the priority ranking is a dynamic ranking that varies according to operations or speed settings of the peripheral devices.

19. (original) The electronic system of claim 11, wherein the host determines which peripheral devices are coupled to the controller and builds a set of virtual drives in an operating system (OS) of the host corresponding the peripheral devices coupled to the controller.

20. (currently amended) A method for accessing a plurality of peripheral devices from a host, the method comprising:

coupling a controller to the host though a single port of a predetermined interconnection means, the ~~single port~~ predetermined interconnection means being designed for providing the host access to a maximum of N devices;

coupling M peripheral devices to the controller, wherein M is greater than N;

modifying predetermined existing fields in packets or registers in an existing task file that are sent to the controller through the single port to specify a target peripheral device; and

accessing the peripheral devices using the single port.

21. (original) The method of claim 20, wherein the predetermined interconnection means is an Integrated Drive Electronics (IDE) bus or a Serial AT Attachment (SATA) interface.

22. (currently amended) The method of claim 20, ~~further comprising:~~

~~modifying predetermined fields in packets or registers in~~ wherein the existing task file is an existing IDE task ~~file~~ file, ~~being sent to the controller through the single port; and~~ the method further comprising determining a target peripheral device according to the predetermined existing fields or the registers.

23. (currently amended) The method of claim ~~22~~ 20, wherein the predetermined fields are control codes or reserved vendor-specific bits in ATA Packet Interface (ATAPI) packets that are sent to the controller through the single port to specify the target peripheral device.

24. (original) The method of claim 20, wherein the M peripheral devices coupled to the controller at least comprise an optical storage device and a non-volatile storage device.

25. (previously presented) The method of claim 24, wherein the non-volatile storage device is a flash card access device or a hard-disk drive.

26. (original) The method of claim 20, further comprising scheduling packets sent to the M peripheral devices according to a priority ranking.

27. (original) The method of claim 26, further comprising dynamically varying the priority ranking according to operations or speed settings of the peripheral devices.

28. (original) The method of claim 20, wherein the M peripheral devices include a first

peripheral device and a second peripheral device, the method further comprising:

directly transferring data stored on the first peripheral device to the second peripheral device without buffering the data in the host.

29. (original) The method of claim 20, further comprising:

determining which peripheral devices are coupled the controller; and

building a set of virtual drives in an operating system (OS) of the host corresponding the peripheral devices coupled to the controller.

30. (currently amended) A method for accessing a plurality of peripheral devices from a host, the method comprising:

coupling a controller to the host though a single port of a predetermined interconnection means, the ~~single port~~ predetermined interconnection means being designed for providing the host access to a maximum of N devices;

coupling M peripheral devices to the controller, wherein M is greater than N and the M peripheral devices include a first peripheral device and a second peripheral device;

modifying predetermined existing fields in packets or registers in an existing task file that are sent to the controller through the single port to specify a target peripheral device;

accessing the peripheral devices using the single ~~port~~ port; and

directly transferring data stored on the first peripheral device to the second peripheral device without buffering the data in the host.

31. (original) The method of claim 30, wherein the predetermined interconnection means is an Integrated Drive Electronics (IDE) bus or a Serial AT Attachment (SATA) interface.

32. (currently amended) The method of claim 30, ~~further comprising:~~

~~modifying predetermined fields in packets or registers in~~ wherein the existing task file is an existing IDE task file, ~~being sent to the controller through the single port; and~~ the method further comprising determining the target peripheral device according to the predetermined existing fields or the registers.

33. (currently amended) The method of claim ~~32~~ 30, wherein the predetermined fields are control codes or reserved vendor-specific bits in ATA Packet Interface (ATAPI) packets that are sent to the controller through the single port to specify the target peripheral device.

34. (original) The method of claim 30, wherein the M peripheral devices coupled to the controller at least comprise an optical storage device and a non-volatile storage device.

35. (original) The method of claim 34, wherein the non-volatile storage device is a flash card access device or a hard-disk drive.

36. (original) The method of claim 25, further comprising scheduling packets sent to the M peripheral devices according to a priority ranking.

37. (original) The method of claim 36, further comprising dynamically varying the priority

ranking according to operations or speed settings of the peripheral devices.

38. (original) The method of claim 30, further comprising:

determining which peripheral devices are coupled the controller; and

building a set of virtual drives in an operating system (OS) of the host corresponding the peripheral devices coupled to the controller.

39. (currently amended) An electronic system comprising:

a host;

a controller electrically coupled to the host to communicate data through a single port of a predetermined interconnection means, the ~~single port~~ predetermined interconnection means being designed for providing the host access to a maximum of N devices;

M peripheral devices electrically coupled to the controller, wherein M is greater than N and the controller allows the host to access the peripheral devices using the single port and the host modifies predetermined existing fields in packets or registers in an existing task file that are sent to the controller through the single port to specify a target peripheral device; and

a memory for storing the data, wherein the memory is shared by the extra (M-N) devices.

40. (previously presented) The electronic system of claim 39, wherein the predetermined interconnection means is an Integrated Drive Electronics (IDE) bus or a Serial AT

Attachment (SATA) interface.

41. (currently amended) The electronic system of claim 39, wherein the ~~host modifies predetermined fields in packets or registers in~~ existing task file is an existing IDE task file ~~that are sent to the controller through the single port to specify a target peripheral device~~.

42. (currently amended) The electronic system of claim ~~41~~ 39, wherein the predetermined fields are control codes or reserved vendor-specific bits in ATA Packet Interface (ATAPI) packets that are sent to the controller through the single port to specify the target peripheral device.

43. (previously presented) The electronic system of claim 39, wherein the M peripheral devices electrically coupled to the controller at least comprise an optical storage device and a non-volatile storage device.

44. (previously presented) The electronic system of claim 43, wherein the non-volatile storage device is a flash card access device or a hard-disk drive.

45. (previously presented) The electronic system of claim 39, wherein the host schedules packets sent to the M peripheral devices according to a priority ranking.

46. (previously presented) The electronic system of claim 45, wherein the priority ranking is a dynamic ranking that varies according to operations or speed settings of the peripheral devices.

47. (previously presented) The electronic system of claim 39, wherein the M peripheral devices include a first peripheral device and a second peripheral device, and the

controller directly transfers data stored on the first peripheral device to the second peripheral device without buffering the data in the host.

48. (previously presented) The electronic system of claim 39, wherein the host determines which peripheral devices are coupled to the controller and builds a set of virtual drives in an operating system (OS) of the host corresponding the peripheral devices coupled to the controller.